**Bài 14: Analyzing Time Series Data phần 2 - Lập trình AI bằng Python**

**5. Trích xuất thống kê từ Time Series Data :**

**a. Trung bình :**

Sử dụng hàm mean() :

**timeseries**.mean()

Kết quả như sau :

-0.11143128165238671

**b. Maximum :**

Sử dụng hàm max :

timeseries.max()

Kết quả :

3.4952999999999999

**c. Minimum :**

Sử dụng hàm min():

timeseries.min()

Kết quả :

-4.2656999999999998

**d. Tổng hợp :**

Nếu bạn muốn tính toán tất cả các thống kê cùng một lúc, sử dụng hàm describe()

**timeseries**.describe()

Kết quả như sau :

count 817.000000

mean -0.111431

std 1.003151

min -4.265700

25% -0.649430

50% -0.042744

75% 0.475720

max 3.495300

dtype: float64

**e. Re-sampling :**

Bạn có thể lấy mẫu lại dữ liệu theo một tần suất thời gian khác. Hai tham số để thực hiện lấy mẫu lại là

* Time period
* Method

**f. Re-sampling với mean()**

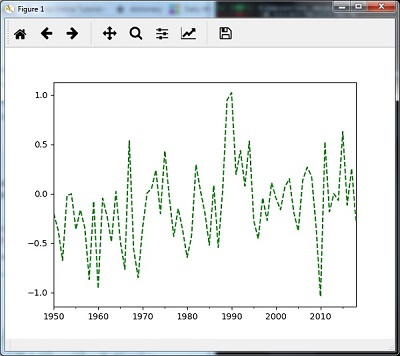
Sử dụng hàm mean() để resample dữ liệu :

**timeseries\_mm** = timeseries.resample("A").mean()

timeseries\_mm.plot(style = 'g--')

plt.show()

Sau đó, bạn có thể quan sát biểu đồ sau là đầu ra của việc lấy mẫu lại bằng cách sử dụng mean ()



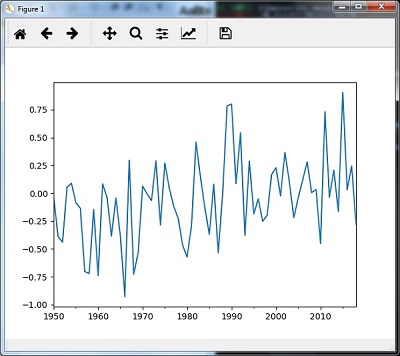
**g. Re-sampling với median()**

**timeseries\_mm** = timeseries.resample("A").median()

timeseries\_mm.plot()

plt.show()

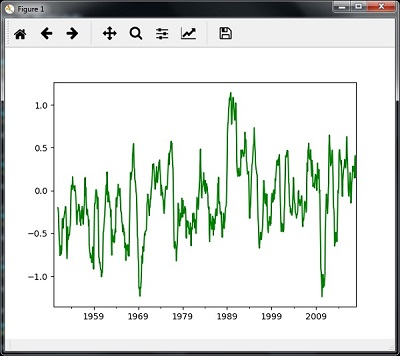
quan sát biểu đồ output :



**i. Rolling Mean**

**timeseries**.rolling(window = 12, center = False).mean().plot(style = '-g')

**plt**.show()



**6. Phân tích dữ liệu tuần tự bằng mô hình Markov ẩn (HMM)**

HMM là một mô hình thống kê được sử dụng rộng rãi cho dữ liệu có tính liên tục và khả năng mở rộng như phân tích thị trường chứng khoán theo chuỗi thời gian, kiểm tra sức khỏe và nhận dạng giọng nói. Phần này đề cập chi tiết đến việc phân tích dữ liệu tuần tự bằng Mô hình Markov ẩn (HMM).

**a. Hidden Markov Model (HMM)**

HMM là một mô hình ngẫu nhiên được xây dựng dựa trên khái niệm chuỗi Markov dựa trên giả định rằng xác suất của các thống kê trong tương lai chỉ phụ thuộc vào trạng thái quy trình hiện tại thay vì bất kỳ trạng thái nào trước đó. Ví dụ, khi tung một đồng xu, chúng ta không thể nói rằng kết quả của lần tung thứ năm sẽ là mặt sấp. Điều này là do một đồng xu không có bất kỳ bộ nhớ nào và kết quả tiếp theo không phụ thuộc vào kết quả trước đó.

Về mặt toán học, HMM bao gồm các biến sau:

**States(S) :**

Là một tập hợp các trạng thái ẩn hoặc tiềm ẩn trong HMM. Nó được ký hiệu là S.

**Output symbols (O) :**

Là một tập hợp các ký hiệu đầu ra có thể có trong HMM. Nó được ký hiệu là O.

**Ma trận xác suất chuyển trạng thái (A) :**

Nó là xác suất thực hiện chuyển đổi từ trạng thái này sang từng trạng thái khác. Nó được ký hiệu là A.

**Ma trận xác suất phát thải quan sát (B) :**

Là xác suất emitting/observing một kí hiệu ở một trạng thái cụ thể. Nó được ký hiệu là B.

**7. Ví dụ :**

Trong ví dụ này, chúng ta sẽ phân tích từng bước dữ liệu của thị trường chứng khoán, để có được ý tưởng về cách HMM hoạt động với dữ liệu tuần tự hoặc chuỗi thời gian. Xin lưu ý rằng chúng tôi đang triển khai ví dụ này bằng Python.

Sử dụng package **matpotlib.finance :**

**import** numpy **as** np

**from** matplotlib **import** cm, pyplot **as** plt

**from** matplotlib.dates **import** YearLocator, MonthLocator

**try**:

**from** matplotlib.finance **import** quotes\_historical\_yahoo\_och1

**except** ImportError:

**from** matplotlib.finance **import** (

quotes\_historical\_yahoo **as** quotes\_historical\_yahoo\_och1)

**from** hmmlearn.hmm **import** GaussianHMM

Load dữ liệu từ start date và end date :

start\_date = datetime.date(1995, 10, 10)

end\_date = datetime.date(2015, 4, 25)

quotes = quotes\_historical\_yahoo\_och1('INTC', start\_date, end\_date)

Trong bước này, ta sẽ trích xuất các báo giá kết thúc mỗi ngày. Đối với điều này, hãy sử dụng lệnh sau:

closing\_quotes = np.array([quote[2] for quote in quotes])

Bây giờ, tôi sẽ trích xuất khối lượng cổ phiếu được giao dịch mỗi ngày. Đối với điều này, hãy sử dụng lệnh sau:

volumes = np.array([quote[5] for quote in quotes])[1:]

Tại đây, lấy phần trăm chênh lệch của giá cổ phiếu đóng cửa, sử dụng đoạn code sau :

diff\_percentages = 100.0 \* np.diff(closing\_quotes) / closing\_quotes[:-]

dates = np.array([quote[0] for quote in quotes], dtype = np.int)[1:]

training\_data = np.column\_stack([diff\_percentages, volumes])

Khởi tạo và huấn luyện Gaussian HMM

hmm = GaussianHMM(n\_components = 7, covariance\_type = 'diag', n\_iter = 1000)

**with** warnings.catch\_warnings():

warnings.simplefilter('ignore')

hmm.fit(training\_data)

Tạo dữ liệu sử dụng mô hình HMM, sử dụng đoạn lệnh sau :

num\_samples = 300

samples, \_ = hmm.sample(num\_samples)

Cuối cùng, trong bước này,mình vẽ và hình dung phần trăm chênh lệch và khối lượng cổ phiếu được giao dịch dưới dạng sản lượng dưới dạng đồ thị.

Sử dụng đoạn code sau để vẽ và hình dung tỷ lệ phần trăm chênh lệch

**plt**.figure()

**plt**.title('Difference percentages')

**plt**.plot(np.arange(num\_samples), samples[:, 0], c = 'black')

Vẽ và hiển thị giá cổ phiếu

**plt**.figure()

**plt**.title('Volume of shares')

**plt**.plot(np.arange(num\_samples), samples[:, 1], c = 'black')

**plt**.ylim(ymin = 0)

**plt**.show()